Suppress hits for KillOnBit18=0 ?

Originally proposed by Jonathan:

- The Mergers downstream the HFs look at the EE from XTFA: if bit 18 is set Merger send out the EE word right away. Afterwards, it processes the hits received from HFs and discards them.

Discussion between Bill, Stefano and myself followed. Two other options emerged.

- Suppress the hits in the HFs.

On each L1A: killonbit18 bit + L2B bits propagate via:

XTFA =>(via bkpl)=> master spy => spy controls =>(via bkpl)=>HFs

HFs read and discard the hits.

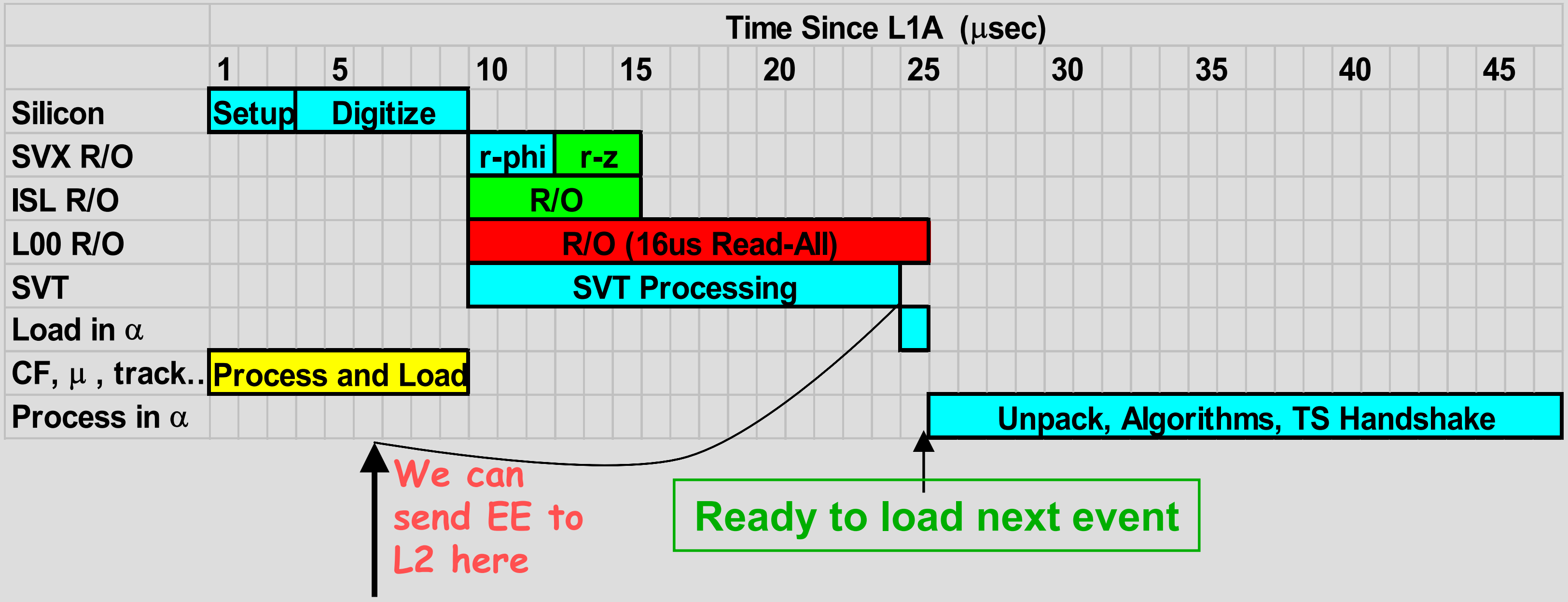
- If the idea is to bypass SVT, lets do it from XTFA to GB
  - ✓Via a Merger: XTFB output is sent to GB second input.
  - ✓GB uses only the EE of the second input to decide if he has to process or discard the event.
    i.e. GB reads the EE of the stream from XTFB (SVT bypass) and check bit 18. If bit18=0:
    - Send out the EE word asap
      - the corresponding event coming to the other (standard) GB input must be skipped.

General question:

Is it worth?

SVT timing decreased by ~1usec with KillOnBit18.

But from the general point of view, it seems an interesting feature to implement:



We can send EE to L2 here

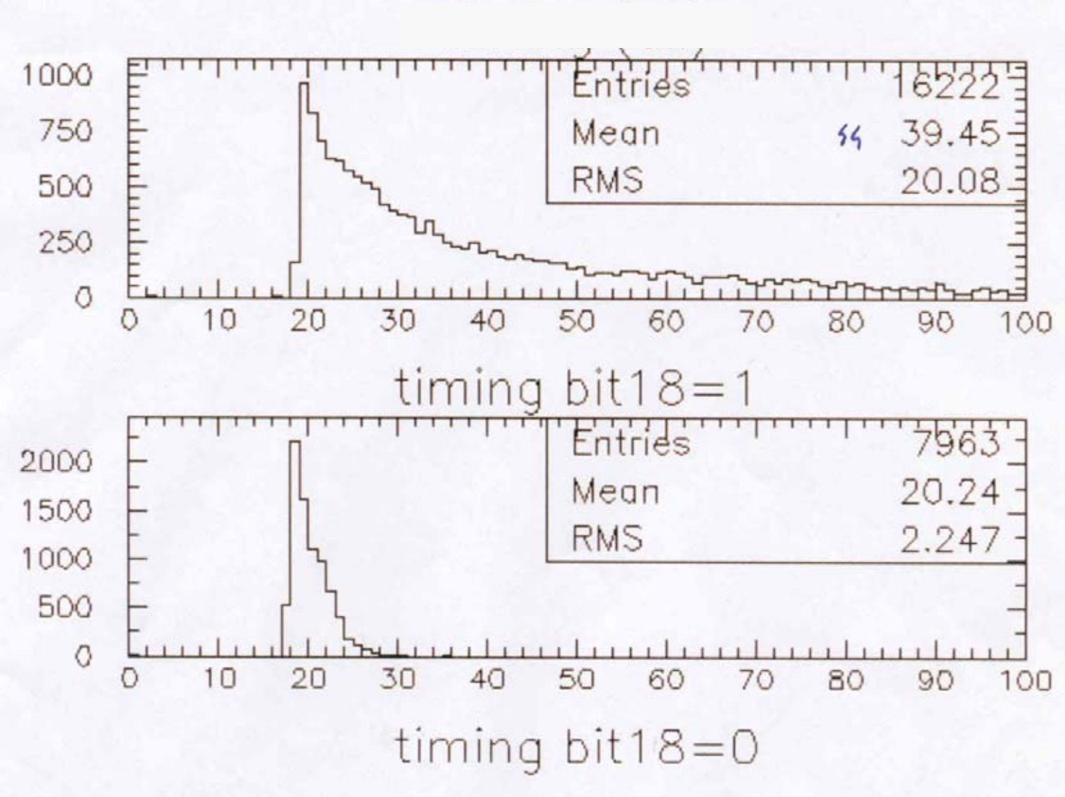
Ready to load next event

~1 year ago diagram

It seems we gain a lot.

At this point I do not fully understand why we don't gain more with implementing KillOnBit18:

- may be will see the effect at high luminosity and with different composition of the trigger



Gain:
1usec

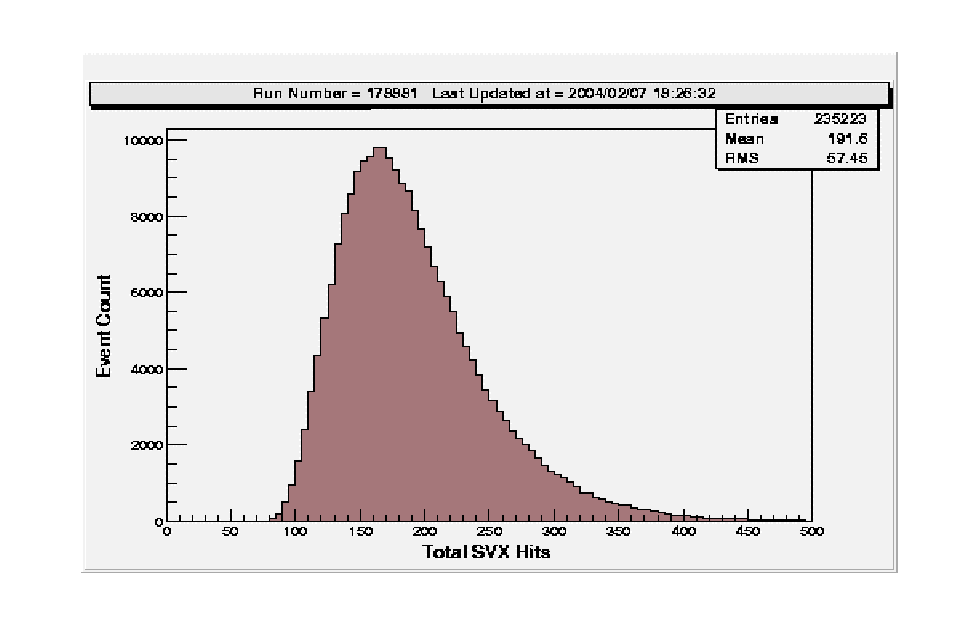
- HF option: seems the nicest but require modifications in XTFA, Master Spy, Spy control, HFs:
  - I doubt we can implement this unless highly motivated
- Merger or GB option:
  - only one board to modify
  - Probably very similar: few hundred nsec (+1usec?)

  Slower with the Merger option but I don't think it matters. It will be very fast anyway.

Merger or GB option:

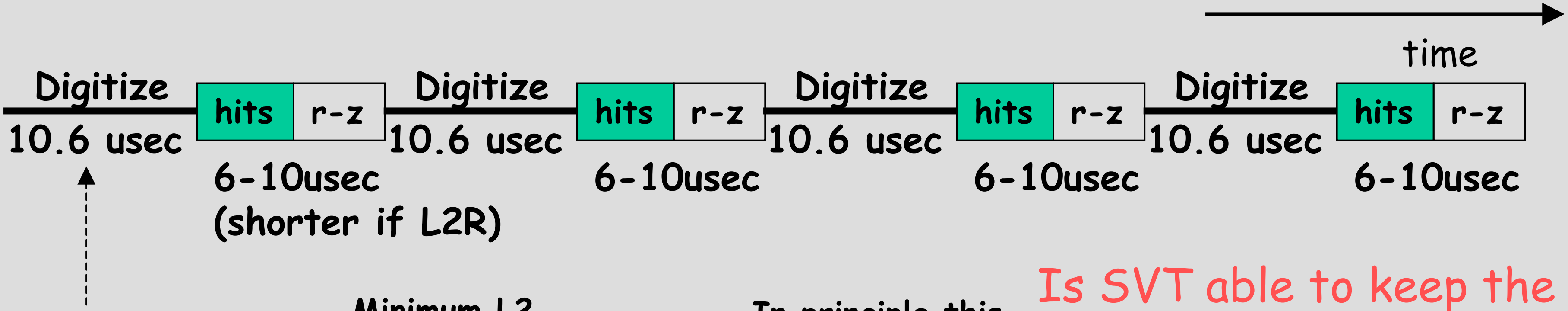
- ➢To me it is highly disturbing to send to level 2 the message that SVT is done while we are still processing data:
  - ➢Is it a real problem? We can handle 4 L1 accept in a raw; but what happens afterwards?
    - ✓Can SVT get a Fifo overflow and completly loose syncronization?

      Are we able to keep the rate?



	200 hits T(usec)	500 hits T(usec)
Merger 33MHz	6	15
HB 23MHz (44ns)	8.8	22

I expect SVX data to arrive at SVT with this pattern (4 L1A in a raw):



time

Digitize 10.6 usec | hits | r-z | 6-10usec (shorter if L2R)

Digitize 10.6 usec | hits | r-z | 6-10usec

Digitize 10.6 usec | hits | r-z | 6-10usec

Digitize 10.6 usec | hits | r-z | 6-10usec

SVT could send the EE to L2 + Minimum L2 processing time: 5-10usec

(from S.Miller)

In principle this pattern can go on with no extra delays

Is SVT able to keep the rate?
Do we feel safe?

Some Doubts

- Is my picture of L2 timing ok? Is it more complicated and there are additional timings entering in the game?
- Do we understand the present effect of KillOnBit18 as implemented right now?
- Should we ask for a simulation of the trigger when SVT suppresses the hits or the behaviour is clear enough?

Conclusions

- I do not like the idea to suppress the hits in Merger or GB. If we do it, I think we should implement the check of the fifo overflows at the Mergers and send a CDF_ERROR to trigger an HRR.
- If it is very important to suppress the hits (as it seems from my simple picture) we should investigate more in detail if the modification of HF (and other boards) is pursuable.

Merger or GB option:

- To me it is highly disturbing to send to level 2 the message that SVT is done while we are still processing data:
    - Is it a real problem? We can handle 4 L1 accept in a raw. When the 4 L2 buffers are full, what is the minimum necessary time to receive the next L1 accept?
        - ✓Can SVT get a Fifo overflow and completly loose syncronization?

          Are we able to keep the rate?